UNITED STATES PATENT APPLICATION

FOR

DYNAMIC CHAIN CREATION AND SEGMENTATION OF THE PACKET-FORWARDING PLANE

INVENTORS:

RAJIV GOEL
JIANYU CHEN
SCOTT MOLLOY
CHUNG T. NGUYEN
DAVID WARD
JOHN BETTINK
PERAMANAYAGAM MARIMUTHU

PREPARED BY:

HICKMAN PALERMO TRUONG & BECKER LLP
1600 WILLOW STREET
SAN JOSE, CA 95125
(408) 414-1080

EXPRESS MAIL MAILING INFORMATION

"Express Mail" mailing label number: EV323352366US

Date of Deposit: April 14, 2004

DYNAMIC CHAIN CREATION AND SEGMENTATION OF THE PACKET-FORWARDING PLANE

FIELD OF THE INVENTION

[0001] The present invention generally relates to routing devices in computer networks. The invention relates more specifically to a method and apparatus for dynamically creating encapsulation and decapsulation chains and segmenting the packet-forwarding plane.

BACKGROUND OF THE INVENTION

[0002] The approaches described in this section could be pursued, but are not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, the approaches described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

[0003] When information is to be transmitted over a computer network, such as a Local Area Network (LAN) or a Wide Area Network (WAN), the information may be inserted, or "encapsulated" into data packets, which are then forwarded, network element-by-network element, from source to destination. Encapsulation typically involves the addition of one or more headers to a data packet that contains a message or data. Each header may contain additional information about how the data packet is to be handled as it traverses a network.

[0004] Multiple layers of encapsulation may be employed when encapsulating information for transmission over a network. Each layer of encapsulation may correspond to a different protocol. For example, an application level protocol header such as a Hypertext Transfer Protocol (HTTP) header may be "prepended" to (i.e., added to the front of) a message that is to be transmitted to an application that uses HTTP. Next, a transport layer

protocol header such as a Transport Control Protocol (TCP) header may be prepended to the HTTP header if the data packet is to be transmitted over a network that uses TCP. Then, a network layer protocol header such as an Internet Protocol (IP) header may be prepended to the TCP header if the data packet is to be transmitted over a network that uses IP. Finally, a data-link protocol header such as an Ethernet Protocol header may be prepended to the IP header if the data packet is to be transmitted over a network that uses the Ethernet Protocol.

[0005] Thus, a message may be prepended with multiple headers during the encapsulation process. The data packet, including the headers, may be forwarded from one network element or forwarding device to another network element or forwarding device. For example, a router may forward a data packet to another router. After a router receives a data packet, the router inspects the contents of the data packet's front-most header. The router may handle the data packet based on the contents of the header. Next, the router may inspect the contents of the data packet's next-to-front-most header, and handle the data packet based on the contents of that header. The router may inspect the contents of each header from the front-most header to the rear-most header in succession, and handle the data packet based on the contents of each such header.

[0006] In handling a data packet, a router may strip a header off of the front of the data packet. For example, if a data packet is to be forwarded through a network that uses the High-level Data Link Control (HDLC) Protocol rather than the Ethernet Protocol, the router may strip the Ethernet Protocol header off of the front of the data packet, so that the HDLC Protocol can be prepended to the data packet instead. The process of inspecting headers as described above, including the possible stripping of such headers, may be called "decapsulation."

[0007] A router typically comprises multiple physical interfaces through which the router receives incoming data packets, and through which the router sends outgoing data packets. Different physical interfaces may be configured to send and/or receive different kinds of data packets. For example, a physical interface might be configured to send and receive only IP Version 4 (IPv4) packets. For another example, a physical interface might be configured to send and receive only IP Version 6 (IPv6) packets. A physical interface could be configured to send and receive both IPv4 and IPv6 packets.

[0008] As described in U.S. Patent No. 6,601,106 B1, each physical interface may be associated with a separate "encapsulation chain" and a separate "decapsulation chain." Each decapsulation chain comprises one or more successive chain elements that successively perform decapsulation functions on data packets as those data packets are passed through those chain elements. Each encapsulation chain comprises one or more successive chain elements that successively perform encapsulation functions on data packets as those data packets are passed through those chain elements. Each physical interface is associated with both an encapsulation chain and a decapsulation chain. Each physical interface is bi-directional.

[0009] A router may receive a data packet on a first of several physical interfaces. The router may pass the data packet through one or more chain elements of the decapsulation chain associated with the first physical interface. At some point during or following the data packet's progression through the first physical interface's decapsulation chain, the router may select, from among the router's multiple physical interfaces, a second physical interface through which the data packet needs to be transmitted in order to move the data packet towards the data packet's ultimate destination. Having made this determination, the router may provide the data packet to a selected chain element in the encapsulation chain that is

associated with the second physical interface. The router may pass the data packet through one or more chain elements of the second physical interface's encapsulation chain. After emerging from the second physical interface's encapsulation chain's last chain element, the data packet may be transmitted out of the router through the second physical interface.

[0010] Formerly, all of a router's physical interfaces were consolidated on a single hardware "card". However, modern distributed routers may comprise multiple separate interconnected cards, such as line cards or routing processors. Each such card contains separate processing and memory resources. Each such card may expose a separate subset of a router's physical interfaces. For each physical interface, the encapsulation and decapsulation chains associated with that physical interface are constructed on the same card that exposes that physical interface. A data packet may be forwarded from a decapsulation chain on a first card to an encapsulation chain on a second card. Thus, a data packet may be received on one of a first card's physical interfaces, and transmitted out on one of a second card's physical interfaces.

[0011] In addition to the physical interfaces described above, a router may comprise one or more virtual interfaces. None of a router's physical ports is a virtual interface per se. Virtual interfaces are embodied in data structures and other software elements, and receive data packets from chains that are associated with physical interfaces. An example of a virtual interface is a "tunnel" interface, which is an interface to functionality that encapsulates a data packet, which conforms to one protocol, into another data packet, which may conform to a different protocol. For example, a tunnel interface may be an interface to functionality that encapsulates an IPv4 packet into an IPv6 packet. For another example, a tunnel interface may be an interface to functionality that encapsulates an IPv6 packet into an IPv4 packet. A tunnel interface may encapsulate an IPv4 packet into an IPv4 packet, or an IPv6 packet into

an IPv6 packet, or a Connectionless Network Service (CLNS) packet into an IPv4 packet, etc.

[0012] As is explained above, a data packet may be received on any one of a distributed router's cards' physical interfaces. According to one approach, in order to allow a data packet to be forwarded to a particular type of virtual interface regardless of which card's physical interface received the data packet, a separate virtual interface of the particular type is provided for each separate card. Each such virtual interface is associated with a separate pair of encapsulation and decapsulation chains. For example, given five separate cards, each card might provide a separate IPv4-to-IPv6 tunnel interface, and each card might implement a separate encapsulation chain and a separate decapsulation chain for each such IPv4-to-IPv6 tunnel interface, resulting in five separate IPv4-to-IPv6 encapsulation chains, and five separate IPv4-to-IPv6 decapsulation chains.

[0013] Unfortunately, under this approach, the encapsulation chains constructed for a particular type of virtual interface on some cards might never be used. For example, given five separate cards, there might not be any physical interfaces on the first card that are configured to send IPv6 data packets. Under these circumstances, the encapsulation chain associated with the IPv4-to-IPv6 tunnel interface for the first card would never be used.

[0014] Similarly, under this approach, the decapsulation chains constructed for a particular type of virtual interface for some cards might never be used. For example, given five separate cards, none of the physical interfaces on the second card might be configured to receive IPv4 data packets. Under these circumstances, the decapsulation chain associated with the IPv4-to-IPv6 tunnel interface for the second card would never be used.

[0015] Encapsulation and decapsulation chains use a card's limited memory and processing resources. Creating chains that will never be used wastes these limited resources, which might otherwise be used for other purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0017] FIG. 1A is a block diagram that illustrates an overview of one embodiment of a data packet-forwarding device in which encapsulation chains and decapsulation chains are selectively created on the device's cards;

[0018] FIG. 1B is a block diagram that illustrates an overview of another embodiment of a data packet-forwarding device in which encapsulation chains and decapsulation chains are selectively created on the device's cards;

[0019] FIG. 2 is a flow diagram that illustrates a high level overview of one embodiment of a method for dynamically and selectively creating encapsulation and decapsulation chains for virtual interfaces, thus segmenting the packet-forwarding plane;

[0020] FIG. 3 depicts a flow diagram that illustrates one embodiment of a method for selectively creating an encapsulation chain and/or a decapsulation chain for a virtual interface; and

[0021] FIG. 4 is a block diagram that illustrates a computer system upon which an embodiment may be implemented.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0022] A method and apparatus for dynamically creating encapsulation and decapsulation chains and segmenting the packet-forwarding plane is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

[0023] The contents of U.S. Patent No. 6,601,106 B1, entitled "PACKET PROCESSING USING NON-SEQUENTIAL ENCAPSULATION AND DECAPSULATION CHAINS" are hereby incorporated by reference in their entirety for all purposes as if fully set forth herein.

[0024] Embodiments are described herein according to the following outline:

1.0 GENERAL OVERVIEW

[0025] The needs identified in the foregoing Background, and other needs and objects that will become apparent for the following description, are achieved in the present invention, which comprises, in one aspect, a method for dynamically creating encapsulation and decapsulation chains and segmenting the packet-forwarding plane. Instead of always creating both an encapsulation and a decapsulation chain for every virtual interface on each

of a router's cards, it is dynamically determined, for each virtual interface of each card, whether both of those chains, only one of those chains, or neither of those chains are useful for that virtual interface of that card. Only those chains that are determined to be useful for that virtual interface are dynamically created on that card (e.g., by creating chain elements in the memory resources of that card). Chains that would be useless for a virtual interface of a card are not created on that card, thereby allowing a portion of that card's resources to be used for other purposes. Thus, a card might provide an encapsulation chain for a virtual interface without providing a decapsulation chain for that virtual interface, or a card might provide a decapsulation chain for a virtual interface without providing an encapsulation chain for that virtual interface, or a card might not provide either an encapsulation chain or a decapsulation chain for that virtual interface. Because either one or the other, both, or neither of an encapsulation chain and a decapsulation chain may be created for a virtual interface of a particular card, it may be said that the packet-forwarding plane has been segmented into independent encapsulation and decapsulation segments.

[0026] In other aspects, the invention encompasses a computer apparatus and a computer-readable medium configured to carry out the foregoing steps.

2.0 STRUCTURAL AND FUNCTIONAL OVERVIEW

[0027] FIG. 1A is a block diagram that illustrates an overview of one embodiment of a data packet-forwarding device 100A in which encapsulation chains and decapsulation chains are selectively created on cards. Data packet-forwarding device 100A may be, for example, a distributed router. Data packet-forwarding device 100A comprises cards 102A-102C and control plane 110.

[0028] Cards 102A-102C are considered to be within the data "plane" of device 100A, as contrasted to control plane 110. Control plane 110 is an abstract representation of one or more processors, devices, and/or software and/or firmware elements. Cards 102A-102C may be, for example, line cards or routing processors. Although three cards are illustrated for purposes of example, data packet-forwarding device according to embodiments of the invention may comprise a greater or lesser number of cards than those illustrated. Each of cards 102A-102C comprises separate memory resources and processing resources. Each of cards 102A-102C is communicatively coupled to control plane 110. Thus, control plane 110 may configure the resources of cards 102A-102C to implement encapsulation and/or decapsulation chains on cards 102A-102C. There may or may not be a control plane presence on cards 102A-102C.

[0029] Each of cards 102A-102C exposes one or more bi-directional physical interfaces. More specifically, in this example, card 102A comprises physical interface 104A; card 102B comprises physical interface 104B; and card 102C comprises physical interface 104C. Although each of cards 102A-102C is illustrated as having one physical interface for purposes of example, cards according to embodiments of the invention may comprise a greater number of physical interfaces than those illustrated.

[0030] Each of physical interfaces 104A-104C may be communicatively coupled to a computer network, such as a LAN, WAN, or internetwork, through which that physical interface sends and/or receives data packets. Each of physical interfaces 104A-104C may be communicatively coupled to a separate network, or to different devices within the same LAN. Each of physical interfaces 104A-104C may be configured to send and/or receive data packets that are structured according to one or more specified protocols. For example, physical interface 104A might be configured to send and receive both IPv4 packets and IPv6

packets; physical interface 104B might be configured to send and receive IPv4 packets but not IPv6 data packets; and physical interface 104C might be configured to send and receive IPv6 packets but not IPv4 data packets.

[0031] Each of cards 102A-102C provides one or more encapsulation and/or decapsulation chains. More specifically, card 102A provides decapsulation chains 106A and 106D and encapsulation chains 108A and 108D; card 102B provides decapsulation chains 106B and 106E and encapsulation chain 106E; and card 102C provides decapsulation chain 106C and encapsulation chains 108C and 108E.

[0032] Each of decapsulation chains 106A-106E and encapsulation chains 108A-108E comprises one or more chain elements as described in U.S. Patent No. 6,601,106 B1. Each such chain element may perform a different encapsulation or decapsulation function. For example, one chain element may decrypt data, another may decompress data, another may perform a switching function, another may rewrite data, another may compress data, and another may encrypt data. Different chains may comprise different chain elements. After processing data, a chain element typically passes the processed data to a next chain element in a chain.

[0033] Physical interface 104A is communicatively coupled to decapsulation chain 106A, which is communicatively coupled to encapsulation chain 108A, which is also communicatively coupled to physical interface 104A. Thus, physical interface 104A may receive a data packet and pass the data packet to decapsulation chain 106A. One or more chain elements in decapsulation chain 106A may process the data packet and pass the data packet to encapsulation chain 108A. One or more chain elements in encapsulation chain 108A may further process the data packet and pass the data packet to physical interface 104A. Physical interface 104A may transmit the data packet out of data packet-forwarding

device 100A. According to one embodiment, any chain element in decapsulation chain 106A may pass a data packet to any chain element in encapsulation chain 108A. Thus, the decapsulation and encapsulation chains may be non-sequential.

[0034] In a similar manner, physical interface 104B is communicatively coupled to decapsulation chain 106B, which is communicatively coupled to encapsulation chain 108B, which is also communicatively coupled to physical interface 104B. In like manner, physical interface 104C is communicatively coupled to decapsulation chain 106C, which is communicatively coupled to encapsulation chain 108C, which is also communicatively coupled to physical interface 104C.

[0035] Because decapsulation chain 106A is communicatively coupled to physical interface 104A, decapsulation chain 106A corresponds to a physical interface. Similarly, decapsulation chain 104B, coupled to physical interface 104A, also corresponds to a physical interface. In contrast, decapsulation chain 106D and encapsulation chain 108D are not communicatively coupled to any physical interface in a direct and immediate sense. Instead, decapsulation chain 106D and encapsulation chain 108D are communicatively coupled to decapsulation chain 106A and encapsulation chain 108A, respectively. Therefore, decapsulation chain 106D and encapsulation chain 108D correspond to virtual interfaces rather than physical interfaces. These virtual interfaces have no direct correspondence to any of physical interfaces 104A-104C.

[0036] A chain element in decapsulation chain 106A may determine that a particular data packet should be sent to a virtual interface that corresponds to one or more chains that provide specialized functionality. For example, a chain element in decapsulation chain 106A might determine, based on an IPv4 packet's destination, that the IPv4 packet should be encapsulated within an IPv6 packet before being transmitted out of data packet-forwarding

device 100A. Due to this determination, the chain element might forward the IPv4 packet to decapsulation chain 106D, which, together with encapsulation chain 108D or encapsulation chain 108E, provides the functionality necessary to encapsulate the IPv4 packet within an IPv6 packet. Although IPv4-to-IPv6 tunneling functionality is used in the examples below, virtual interfaces according to embodiments of the invention may provide interfaces to various other functionalities, such as Generic Routing Encapsulation (GRE) tunneling functionality, MPLS Traffic Engineering and IP Security Protocol (IPsec) functionality.

[0037] Decapsulation chain 106D may process the data packet and pass the data packet to encapsulation chain 108D or encapsulation chain 108E, depending on whether the data packet is supposed to be transmitted out on physical interface 104A or physical interface 104C. Encapsulation chain 108D or encapsulation chain 108E may further process the packet to encapsulate the IPv4 packet into an IPv6 packet. The IPv6 packet may then be passed to encapsulation chain 108A, if encapsulation chain 108D processed the data packet, or to encapsulation chain 108C, if encapsulation chain 108E processed the data packet.

[0038] Thus, decapsulation chains 106D and 106E and encapsulation chains 108D and 108E are associated with virtual interfaces that provide IPv4-to-IPv6 functionality. A separate IPv4-to-IPv6 virtual interface is provided by each of cards 102A-102C. However, not every IPv4-to-IPv6 virtual interface is associated with both an encapsulation chain and a decapsulation chain on the same card as the IPv4-to-IPv6 virtual interface. Although the IPv4-to-IPv6 virtual interface on card 102A is associated with both decapsulation chain 106D and encapsulation chain 108D, the IPv4-to-IPv6 virtual interface on card 102B is associated with only decapsulation chain 106E, and the IPv4-to-IPv6 virtual interface on card 102C is associated with only encapsulation chain 108E. Card 102B does not provide an

encapsulation chain for the IPv4-to-IPv6 virtual interface on card 102B, and card 102C does not provide a decapsulation chain for the IPv4-to-IPv6 virtual interface on card 102C.

[0039] In the example above, because physical interface 104B is not configured to send IPv6 data packets, there is no need to create, on card 102B, an encapsulation chain for an IPv4-to-IPv6 virtual interface. Similarly, because physical interface 104C is not configured to receive IPv4 data packets, there is no need to create, on card 102C, a decapsulation chain for an IPv4-to-IPv6 interface. By avoiding the creation of such an encapsulation chain on card 102B and such a decapsulation chain on card 102C, the resources of cards 102B and 102C are conserved, while still providing an IPv4-to-IPv6 virtual interface on each card. Data packets processed by either decapsulation chain 106D or 106E may be passed to either encapsulation chain 108D or 108E, depending on those data packet's destinations. Encapsulation chains 108D and 108E may receive data packets from both decapsulation chains 106D and 106E.

[0040] According to one embodiment, each of physical interfaces 104A-104C may be associated with one or more destinations that eventually can be reached through that physical interface. Such destinations may be "learned" dynamically and associated with physical interfaces using routing protocols such as Border Gateway Protocol (BGP) and Open Shortest-Path First (OSPF) Interior Gateway Protocol. Destinations may be, for example, IP addresses, Multiprotocol Label Switching (MPLS) labels, networks, etc. Each virtual interface may be associated with a separate destination. According to one embodiment, an encapsulation chain for a particular virtual interface is not created on a card unless that card has at least one physical interface through which the particular virtual interface's associated destination eventually can be reached. According to one embodiment, a decapsulation chain for a particular virtual interface is not created on a card unless that card has at least one

physical interface that is reachable by a network that carries data packets that are destined for the particular virtual interface's associated destination.

[0041] According to one embodiment, no decapsulation chains are created for MPLS Traffic Engineering (TE) virtual interfaces. According to one embodiment, an encapsulation chain for an MPLS TE virtual interface is created only on one card in a routing device: the card that has the physical interface that receives MPLS TE packets (i.e., the card that has the physical interface to which the MPLS TE virtual interface is tied).

[0042] FIG. 1B is a block diagram that illustrates an overview of another embodiment of a data packet-forwarding device 100B in which encapsulation chains and decapsulation chains are selectively created on cards. Data packet-forwarding device 100B may be, for example, a distributed router. Data packet-forwarding device 100B comprises cards 102D-102F and control plane 110. Each of cards 102D-102F is communicatively coupled to control plane 110. Thus, control plane 110 may configure the resources of cards 102D-102F to implement encapsulation and/or decapsulation chains on cards 102D-102F.

[0043] Cards 102E and 102F provide one or more bi-directional physical interfaces. More specifically, card 102E comprises physical interface 104E, and card 102F comprises physical interface 104F. However, card 102D does not provide any physical interfaces. Card 102D is a special-purpose card that provides specialized functionality that may be used by cards 102E and 102F. For example, card 102D may provide IPsec processing functionality.

[0044] On card 102E, physical interface 104E is communicatively coupled to decapsulation chain 106F, which is communicatively coupled to encapsulation chain 108F, which is also communicatively coupled to physical interface 104E. On card 102F, physical interface 104F is communicatively coupled to decapsulation chain 106G, which is

communicatively coupled to encapsulation chain 108G, which is also communicatively coupled to physical interface 104F.

[0045] Decapsulation chains 106F and 106G and encapsulation chains 108F and 108G correspond to physical interfaces. In contrast, decapsulation chain 106H and encapsulation chain 108H, located on card 102D, are not communicatively coupled to any physical interface in a direct and immediate sense. Instead, decapsulation chain 106H is communicatively coupled to decapsulation chains 106F and 106G, and encapsulation chain 108H is communicatively coupled to encapsulation chains 108F and 108G. Decapsulation chain 106H and encapsulation chain 108H correspond to virtual interfaces rather than physical interfaces.

[0046] A chain element in decapsulation chain 106F may determine that a particular data packet should be sent to a virtual interface that corresponds to one or more chains that provide specialized functionality. For example, a chain element in decapsulation chain 106F might determine that a data packet should undergo IPsec processing before being transmitted out of data packet-forwarding device 100B. Due to this determination, the chain element might forward the data packet to decapsulation chain 106H, which, together with encapsulation chain 108H, provides IPsec processing functionality.

[0047] Thus, decapsulation chain 106H and encapsulation chain 108H are associated with virtual interfaces that provide IPsec processing functionality. A separate IPsec virtual interface is provided by each of cards 102E and 102F. However, the IPsec virtual interface for card 102E is not associated with encapsulation and decapsulation chains on the same card as the IPsec virtual interface for card 102E. Likewise, the IPsec virtual interface for card 102F is not associated with encapsulation and decapsulation chains on the same card as the IPsec virtual interface for card 102F. Card 102E does not provide encapsulation or

decapsulation chains for the IPsec virtual interface on card 102E, and card 102F does not provide encapsulation and decapsulation chains for the IPsec virtual interface on card 102F.

[0048] In the example above, because card 102D provides decapsulation chain 106H and encapsulation chain 108H to process data packets according to IPsec, there is no need to create, on cards 102E or 102F, decapsulation or encapsulation chains that perform IPsec processing. Instead of redundantly creating such chains on cards 102E and 102F, control plane 110 may create minimal contexts or states on cards 102E and 102F (e.g., within the memory resources of cards 102E and 102F). These minimal contexts or states may refer or point to decapsulation chain 106H on card 102D. These minimal contexts or states may be associated with the IPsec virtual interfaces of their respective cards.

[0049] By avoiding the creation of IPsec encapsulation and decapsulation chains on cards 102E and 102F, the resources of cards 102E and 102F are conserved, while still providing an IPsec virtual interface on each card. If data packets processed by either decapsulation chain 106F or 106G require IPsec processing, then those data packets may be passed to decapsulation chain 106H. Encapsulation chains 108F and 108G may receive data packets from encapsulation chain 108H.

[0050] FIG. 2 is a flow diagram 200 that illustrates a high level overview of one embodiment of a method for dynamically and selectively creating encapsulation and decapsulation chains, thus segmenting the packet-forwarding plane. In block 202, from among a plurality of cards of a packet-forwarding device, one or more first cards are selected based on some criteria. For example, assuming that the selection is being performed in light of a particular IPv4-to-IPv6 tunnel, control plane 110 might select, from among cards 102A-102C, only cards that have at least one physical interface that is (a) configured or otherwise enabled to send and receive data packets that conform to the IPv6 protocol, and (b)

associated with (i.e., can eventually reach) the particular IPv4-to-IPv6 tunnel's associated destination. For another example, control plane 110 might select, as the one or more first cards, only those of cards 102A-102C that have been specified by first user input.

[0051] In block 204, from among the plurality of cards, one or more second cards are selected based on some criteria. For example, assuming that the selection is being performed in light of a particular IPv4-to-IPv6 virtual interface, control plane 110 might select, from among cards 102A-102C, only cards that have at least one physical interface that is (a) configured or otherwise enabled to send and receive data packets that conform to the IPv4 protocol, and (b) reachable by a network that carries data packets that are destined for the particular virtual interface's associated destination. For another example, control plane 110 might select, as the one or more second cards, only those of cards 102A-102C that have been specified by second user input.

[0052] In block 206, on each of only the one or more first cards, an encapsulation chain is created for a virtual interface for that card. For example, control plane 110 might create, on each of the one or more first cards, a separate encapsulation chain for a particular IPv4-to-IPv6 virtual interface. Thus, control plane 110 would not create, on those of cards 102A-102C that lack physical interfaces that are (a) configured or otherwise enabled to send and receive IPv6 packets and (b) associated with the particular IPv4-to-IPv6 tunnel's associated destination, an encapsulation chain for the particular IPv4-to-IPv6 virtual interface.

[0053] In block 208, on each of only the one or more second cards, a decapsulation chain is created for a virtual interface for that card. For example, control plane 110 might create, on each of the one or more second cards, a separate decapsulation chain for the particular IPv4-to-IPv6 virtual interface. Thus, control plane 110 would not create, on those of cards 102A-102C that lack physical interfaces that are (a) configured to send and receive IPv4 data

packets and (b) reachable by a network that carries data packets that are destined for the particular virtual interface's associated destination, a decapsulation chain for the particular IPv4-to-IPv6 virtual interface. Instead, control plane 110 might create, on those cards, a minimal context that refers to or points to such a decapsulation chain that has been created on another one of cards 102A-102C.

[0054] As a result of the method illustrated in flow diagram 200, data packet-forwarding devices such as those illustrated in FIG. 1A and FIG. 1B may be produced. In such data packet-forwarding devices, a card may provide a virtual interface to specialized functionality without having both (or either) encapsulation and decapsulation chains that correspond to that virtual interface. This beneficially differs from other approaches, in which an encapsulation chain for a virtual interface and a decapsulation chain for the virtual interface both were created on a card even if at least one of those chains was not needed on the card.

[0055] The method illustrated in flow diagram 200 may be performed repetitively. For example, the method may be performed at periodic intervals, or in response to occurrences of specified events. The method may be performed in response to a detected change in network topography, and/or in response to user configuration. The method may be performed in response to the addition or removal of a card from a packet-forwarding device. When a particular chain is no longer needed for a particular virtual interface, then any resources that were formerly used for that chain may be freed and made available for other purposes. Thus, virtual interfaces' encapsulation chains and decapsulation chains may be created and removed dynamically.

[0056] Detailed example implementations of the foregoing general approach are described below.

3.0 IMPLEMENTATION EXAMPLES

[0057] As is described above, a packet-forwarding device may comprise a plurality of cards, and each such card may comprise a plurality of virtual interfaces. According to one embodiment, the technique described below is performed for each virtual interface provided by any of a packet-forwarding device's cards.

[0058] For each such virtual interface, a number of encapsulation chains to be created for that virtual interface is determined: either one or zero. Additionally, for each such virtual interface, a number of decapsulation chains to be created for that virtual interface is determined: either one or zero.

[0059] Once the number of encapsulation chains and the number of decapsulation chains for a particular virtual interface have been determined, then the determined numbers of encapsulation chains and decapsulation chains are created for the virtual interface on the card that provides that virtual interface. In one embodiment, the numbers of encapsulation and decapsulation chains to be created for a particular virtual interface are determined in the manner described below.

[0060] FIG. 3 depicts a flow diagram 300 that illustrates one embodiment of a method for selectively creating an encapsulation chain and/or a decapsulation chain for a particular virtual interface of a particular card within a plurality of cards of a data packet-forwarding device.

[0061] In block 302, it is determined whether the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by or more chains for the particular virtual interface. For example, given an IPsec virtual interface on card 102E, control plane 110 may determine whether any of cards 102D-102F is

a specialized card that is designed to perform IPsec processing on data packets. Chains for an IPsec virtual interface would perform IPsec processing on data packets.

[0062] For another example, given an IPv4-to-IPv6 virtual interface on card 102C, control plane 110 may determine whether any of cards 102A-102C is a specialized card that is designed to encapsulate IPv4 packets within IPv6 packets. Chains for an IPv4-to-IPv6 interface would encapsulate IPv4 packets within IPv6 packets.

[0063] If the plurality of cards includes a specialized card that is designed to perform a type of data packet processing that would be performed by chains for the particular virtual interface, then control passes to block 304. Otherwise, control passes to block 306.

[0064] In block 304, no encapsulation or decapsulations chains are created, on the particular card, for the particular virtual interface. The number of encapsulation chains and the number of decapsulation chains to be created for the particular virtual interface are both selected to be zero. No resources of the particular card are used to create an encapsulation chain for the particular virtual interface, and no resources of the particular card are used to create a decapsulation chain for the particular virtual interface. If any resources of the particular card were being used for an encapsulation chain for the particular virtual interface, then those resources are freed for other purposes. If any resources of the particular card previously were being used for a decapsulation chain for the particular virtual interface, then those resources are freed for other purposes. A minimal context or state, which refers to or points to a chain on the specialized card that was found within the plurality of cards, may be created on the particular card and associated with the particular virtual interface.

[0065] Alternatively, in block 306, it is determined whether at least one physical port of the particular card is (a) configured or otherwise enabled to send data packets of a type that would be produced by an encapsulation chain for the particular virtual interface and (b)

associated with (i.e., can eventually reach) the particular virtual interface's associated destination. For example, given an IPv4-to-IPv6 virtual interface on card 102C, control plane 110 may determine whether physical interface 104C is configured to send and receive IPv6 data packets, which would be produced by an encapsulation chain for an IPv4-to-IPv6 virtual interface, and whether physical interface 104C is associated with the IPv4-to-IPv6 virtual interface's associated destination. If at least one physical port of the particular card is (a) configured to send data packets of the type that would be produced by an encapsulation chain for the particular virtual interface and (b) associated with the particular virtual interface's associated destination, then control passes to block 310. Otherwise, control passes to block 308.

[0066] In block 308, no encapsulation chains are created, on the particular card, for the particular virtual interface. The number of encapsulation chains to be created for the particular virtual interface is selected to be zero. No resources of the particular card are used to create an encapsulation chain for the particular virtual interface. If any resources of the particular card previously were being used for an encapsulation chain for the particular virtual interface, then those resources are freed for other purposes. Control passes to block 312.

[0067] Alternatively, in block 310, an encapsulation chain is created, on the particular card, for the particular virtual interface. The encapsulation chain is associated with the particular virtual interface. The number of encapsulation chains to be created for the particular virtual interface is selected to be one. Resources of the particular card are used to create the encapsulation chain for the particular virtual interface. Control passes to block 312.

[0068] In block 312, it is determined whether at least one physical port of the particular card is configured to receive data packets of a type that would be received by a decapsulation chain for the particular virtual interface. For example, given an IPv4-to-IPv6 virtual interface on card 102B, control plane 110 may determine whether physical interface 104B is (a) configured to send and receive IPv4 data packets, which would be processed by a decapsulation chain for an IPv4-to-IPv6 virtual interface, and (b) reachable by a network that carries data packets that are destined for the particular virtual interface's associated destination. If at least one physical port of the particular card is (a) configured to receive data packets of the type that would be processed by a decapsulation chain for the particular virtual interface and (b) reachable by a network that carries data packets that are destined for the particular virtual interface's associated destination, then control passes to block 316. Otherwise, control passes to block 314.

[0069] In block 314, no decapsulation chains are created, on the particular card, for the particular virtual interface. The number of decapsulation chains to be created for the particular virtual interface is selected to be zero. No resources of the particular card are used to create a decapsulation chain for the particular virtual interface. If any resources of the particular card previously were being used for a decapsulation chain for the particular virtual interface, then those resources are freed for other purposes

[0070] Alternatively, in block 316, a decapsulation chain is created, on the particular card, for the particular virtual interface. The decapsulation chain is associated with the particular virtual interface. The number of decapsulation chains to be created for the particular virtual interface is selected to be one. Resources of the particular card are used to create the decapsulation chain for the particular virtual interface.

[0071] Thus, for any virtual interface on any card of a data packet-forwarding device, one or the other, neither, or both of an encapsulation chain and a decapsulation chain may be created. By allowing less than two chains to be created for virtual interfaces, the resources of the data packet-forwarding device are conserved without loss of encapsulation or decapsulation functionality.

4.0 IMPLEMENTATION MECHANISMS -- HARDWARE OVERVIEW

[0072] FIG. 4 is a block diagram that illustrates a computer system 400 upon which an embodiment of the invention may be implemented. The preferred embodiment is implemented using one or more computer programs running on a network element such as a router device. Thus, in this embodiment, the computer system 400 is a router.

[0073] Computer system 400 includes a bus 402 or other communication mechanism for communicating information, and a processor 404 coupled with bus 402 for processing information. Computer system 400 also includes a main memory 406, such as a random access memory (RAM), flash memory, or other dynamic storage device, coupled to bus 402 for storing information and instructions to be executed by processor 404. Main memory 406 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 404. Computer system 400 further includes a read only memory (ROM) 408 or other static storage device coupled to bus 402 for storing static information and instructions for processor 404. A storage device 410, such as a magnetic disk, flash memory or optical disk, is provided and coupled to bus 402 for storing information and instructions.

[0074] A communication interface 418 may be coupled to bus 402 for communicating information and command selections to processor 404. Interface 418 is a conventional serial

interface such as an RS-232 or RS-422 interface. An external terminal 412 or other computer system connects to the computer system 400 and provides commands to it using the interface 414. Firmware or software running in the computer system 400 provides a terminal interface or character-based command interface so that external commands can be given to the computer system.

[0075] A switching system 416 is coupled to bus 402 and has an input interface 414 and an output interface 419 to one or more external network elements. The external network elements may include a local network 422 coupled to one or more hosts 424, or a global network such as Internet 428 having one or more servers 430. The switching system 416 switches information traffic arriving on input interface 414 to output interface 419 according to pre-determined protocols and conventions that are well known. For example, switching system 416, in cooperation with processor 404, can determine a destination of a packet of data arriving on input interface 414 and send it to the correct destination using output interface 419. The destinations may include host 424, server 430, other end stations, or other routing and switching devices in local network 422 or Internet 428.

[0076] The invention is related to the use of computer system 400 for avoiding the storage of client state on computer system 400. According to one embodiment of the invention, computer system 400 provides for such updating in response to processor 404 executing one or more sequences of one or more instructions contained in main memory 406. Such instructions may be read into main memory 406 from another computer-readable medium, such as storage device 410. Execution of the sequences of instructions contained in main memory 406 causes processor 404 to perform the process steps described herein. One or more processors in a multi-processing arrangement may also be employed to execute the sequences of instructions contained in main memory 406. In alternative embodiments, hard-

wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0077] The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 404 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 410. Volatile media includes dynamic memory, such as main memory 406. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 402. Transmission media can also take the form of acoustic or light waves, such as those generated during radio wave and infrared data communications.

[0078] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punch cards, paper tape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0079] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 404 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 400 can receive the data on the telephone line and use an infrared transmitter to convert the data to an infrared signal. An infrared detector coupled to bus 402 can receive the data carried in the infrared signal and place the data on bus 402. Bus 402 carries the data to main memory 406, from which

processor 404 retrieves and executes the instructions. The instructions received by main memory 406 may optionally be stored on storage device 410 either before or after execution by processor 404.

[0080] Communication interface 418 also provides a two-way data communication coupling to a network link 420 that is connected to a local network 422. For example, communication interface 418 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 418 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 418 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0081] Network link 420 typically provides data communication through one or more networks to other data devices. For example, network link 420 may provide a connection through local network 422 to a host computer 424 or to data equipment operated by an Internet Service Provider (ISP) 426. ISP 426 in turn provides data communication services through the worldwide packet data communication network now commonly referred to as the "Internet" 428. Local network 422 and Internet 428 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 420 and through communication interface 418, which carry the digital data to and from computer system 400, are exemplary forms of carrier waves transporting the information.

[0082] Computer system 400 can send messages and receive data, including program code, through the network(s), network link 420 and communication interface 418. In the

Internet example, a server 430 might transmit a requested code for an application program through Internet 428, ISP 426, local network 422 and communication interface 418. In accordance with the invention, one such downloaded application provides for avoiding the storage of client state on a server as described herein.

[0083] Processor 404 may execute the received code as it is received and/or stored in storage device 410 or other non-volatile storage for later execution. In this manner, computer system 400 may obtain application code in the form of a carrier wave.

5.0 EXTENSIONS AND ALTERNATIVES

[0084] In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.